

# WEB - программирование

## Тема 4 Передача данных на сервер

# Передача данных

- Для перехода на разные части выполнения программы используется метод создания адреса с применением **переменных и их значений**.
- При отправке данных к **URL**-адресу после знака **вопроса** добавляется конструкция в виде пар: **ИМЯ=значения**, объединенных с помощью амперсанта **&**:

# Пример передачи данных

**action**?name1=value1&name2=value2&name3=value3

- Здесь **action** – это **URL-адрес программы** (страницы, с которой передаются данные), которая должна обрабатывать форму (это либо программа, заданная в атрибуте *action* тега *form*, либо сама текущая программа, если этот атрибут отсутствует).
- Имена **name1, name2, name3** соответствуют именам элементов формы (или имена переменных)
- Значения - **value1, value2, value3** – соответствуют значениям этих элементов (или значениям переменных).

# Передача данных

Строка *URL* с переменными и их значениями условно выглядит так:

`http://phpbook.info/test.php?id=10&user=pit`

---

# Передача переменных по ссылке

Ссылка организуется таким образом, что в ее состав, после имени файла, который должен загрузиться, вписываются передаваемые переменные и принимаемые ими значения.

# Передача переменных по ссылке

- **Переменные** отделяются от имени файла знаком - **?**
- **Переменные** отделяются друг от друга знаком - **&**
- Переменные отделяются от своих значений знаком - **=**
- Если переменная несет логический смысл, ее значение может отсутствовать
- Адрес формируется без пробелов и прочих знаков препинания.

# Вид ссылки URL -адрес

Тогда вид ссылки будет следующим:

`<a href="имя файла ? переменная1 = значение1 & переменная2 = значение2 & переменная3 = значение3">Текст ссылки</a>`

Например:

`<a href="index.php?id=about&page=1&ph=5">Текст ссылки</a>`

# Функции для работы с файлами

- Для включения **кода из файла в текст программы** используются функции:

```
include("имя_файла");  
require("имя_файла");
```

Действуют они одинаково, отличие в том, что **include** можно использовать в цикле, чтобы при каждой новой итерации вызывался новый файл.



# Суперглобальный массив

## `$_SERVER[PHP_SELF]`

- **`$_SERVER`** - это суперглобальный массив, содержащий такую информацию, как заголовки, пути, размещение скриптов. Данный массив создается веб-сервером.

# Параметр PHP\_SELF

- **PHP\_SELF** - Имя файла в настоящее время выполняющегося сценария, относительно корня документа.
- Например,  
***`$_SERVER['PHP_SELF']`*** в сценарии при адресе <http://example.com/test.php> возвращает </test.php>.

# Функции для работы с файлами

- Файлы представляют собой обычные
  - текстовые файлы (\*.txt) или
  - файлы формата \*.php, в которых текст раздела можно предварительно **оформить в HTML виде.**

# Функция для загрузки файла

- Для включения кода из файла в текст программы используются функции:
- **include**("имя\_файла");
- **require**("имя\_файла");
- Действуют они одинаково, отличие в том, что **include** можно использовать в цикле, чтобы при каждой новой итерации вызывался новый файл.

---

# ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ

# ФУНКЦИИ ДЛЯ ВЫЧИСЛЕНИЯ

- Синтаксис:

```
function name (arguments) {  
    actions;  
    return variable;  
}
```

# Пример функции файл **f\_sum.php**

```
function summa ($n1,$n2) {  
    $sum=$n1+$n2;  
    return $sum;  
}
```

# Порядок работы с функцией

Файл **example.php**

```
<?php
```

```
$num1=10;
```

```
$num2=12;
```

```
include("f_sum.php");
```

```
$result = summa($num1,$num2);
```

```
echo $result; // на странице вывод 22
```

```
?>
```